UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/677,121	09/30/2003	Andreas Roessler	13913-058001 / 2002P00218	3906

32864 7590 02/07/2008
FISH & RICHARDSON, P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
THERIAULT, STEVEN B	

ART UNIT	PAPER NUMBER
2179	

MAIL DATE	DELIVERY MODE
02/07/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

MN

Office Action Summary	**Application No.** 10/677,121	**Applicant(s)** ROESSLER, ANDREAS	
	Examiner Steven B. Theriault	**Art Unit** 2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 November 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-12,14 and 15* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-12,14 and 15* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the following communications: Amendment 11/14/2007 with an Information disclosure statement

This action is made Final.

2. Claims 1, 3-12, 14-15 are pending in the case. Claims 1, 9 12 and 15 are the independent claims. Claims 2 and 13 have been cancelled.

Specification

3. The amendment to correct a duplicate sentence in Para 12 is accepted and the removal of one word in Para 31 is accepted.

Claim Rejections - 35 USC § 101

4. In light of applicant's arguments and amendments the previous 101 rejection is now considered moot and removed.

Claim Rejections - 35 USC § 103

5. **The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:**

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

6. **Claims 1, 3-12, 14-15 are rejected under 35 U.S.C 103(a) as being unpatentable over Vaidya et al (hereinafter Vaidya) U.S. Publication No. 2002/0083073 published June 27, 2002, in view of Lewallen et al. (Hereinafter Lewallen) U.S. Patent No. 6675230 filed Aug. 22, 2000.**

In regard to **Independent claim 1,** Vaidya teaches a computer program product, tangibly embodied in an information carrier, the computer program product comprising instructions operable to cause data processing apparatus to perform operations comprising:

- Identifying in a web browser on a client computer a UI element being an element of a user interface page received from a server computer, rendered by the browser, and provided as a screen presentation to a user of the client computer, (See Para 93) a change of a user interface (UI) element that references a node of a document object model (DOM) hierarchy (Vaidya Para 0047-0048 and Figure 1 and Para 67) Vaidya teaches that a user operation on a DOM tree node is identified. All elements in a browser are represented in a browser. All browsers parse a document object model to represent the elements in the browser window (See the Document Object Model standard version 1, 1998 (http://www.w3.org/TR/REC-DOM-Level-1/). Vaidya shows the user has identified the need to change information in the browser
- Determining whether the change of the UI element can be applied to the DOM hierarchy by using an update function (Vaidya See Para 0069-0076)
- If the change can be applied by using the update function, finding in the DOM hierarchy the node that is referenced by the UI element and modifying at least one attribute related to the node according to the change by using the update function (Vaidya Para 0076-0077) Vaidya teaches that at least on attribute is changed through the update function for the node.
- Else, setting a dirty flag. (See Para 0062) Vaidya teaches that the replaced state attribute for a given node was not known to the previous layers and is new and will have to be

updated in the cache.

Vaidya does not expressly teach:

- the identified change occurring after the user interface page had been received and rendered by the browser in the client computer; determining in the browser whether the change of the UI element can be applied to the DOM hierarchy by ~~using~~ an ~~update~~ function applying a delta renderer to the DOM hierarchy, the delta renderer being a set of update functions that can modify the DOM representation of a UI element directly in the browser;
  if the change can be applied by the update ~~function~~ delta renderer, the browser finding in the DOM hierarchy the node that is referenced by the UI element and modifying at least one attribute related to the node according to the change by using at least one of the update functions of the delta renderer to directly modify the attribute in the DOM hierarchy;

However, Lewallen teaches a process of accepting and embedding during runtime mixed program statements from a variety of program types at a local client by manipulating the DOM (See column 10, lines 43-67) for the purposes of modifying HTML upon the occurrence of certain types of user input. The client uses a local bridge to access APIs to modify the user interface elements that are rendered on the display. The bridge creates a new interface object and a node for the object which includes a pointed to the object that is changing (See column 7, lines 60-67 and column 8, lines 1-15 and column 11, lines 1-15). Lewallen specifically teaches a process of applying a change (delta renderer) to a document when a condition of a change is to be applied to an interface Element. For example, Lewallen teaches the process of querying the pointer to an interface object. The bridge passes the pointer to the document stub. If the stub reports that the interface element is not supported by the browser then bridge looks for other objects to consider to find a match to the object type. When a match is found then the mixed program statement is executed at runtime and within the browser to render the object (See column 12, lines 44-67 and 1-67). Lewallen further teaches the process of creating a new element by referencing the embedded element in the DOM and a new element is embedded in the DOM with the correct object type (See column 14, lines 29-67). Lewallen and Vaidya teach processes of updating nodes in a document object model. They both teach processes of accessing a given node and element.

Accordingly, it would have been obvious to one of ordinary skill in the art at the time of the invention having the teachings of Vaidya and Lewallen in front of them to modify the system of Vaidya to include the client side manipulation of the node elements by storing a new element as a node in the DOM. The motivation to combine Lewallen with Vaidya comes from the suggestion in Lewallen that many interface programs, such as browsers, use and implement the W3C DOM standard and API's and by providing a mapping between the DOM standard and application object the objects can be manipulated by modifying the DOM (See column 3, lines 50-67 and column 5, lines 10-25).

With respect to **dependent claim 3,** Vaidya teaches the product wherein the dirty flag indicates invalidity of a cached DOM representation of the UI element (See Para 0060) Vaidya teaches the state of the node has changed and therefore the cache and the tree are not is sync and needs to be updated (See also Para 0057-0063).

With respect to **dependent claim 4,** as indicated in the above discussion, Vaidya in view of Lewallen teaches each element of claim 1.

Vaidya does not expressly teach the product wherein the dirty flag indicates invalidity of a cached DOM representation of a further UI element that is a parent of the UI element. However, this limitation would have been obvious to one of ordinary skill in the art at the time of the invention, in view of Lewallen, because Lewallen teaches a process of checking to see if the node type is supported and if not then a new element is created and embedded in the DOM. The process in Lewallen teaches modifying the DOM for the purposes of changing the code statements to run on the native system where the objects were not previously recognized.

With respect to **dependent claim 5,** Vaidya teaches the product wherein the cached DOM representation comprises the node and the at least one attribute is an attribute of the node (See Para 0055 and 0065). Vaidya teaches state attributes as well as data attributes at each node

(See Para 0047).

With respect to **dependent claim 6,** Vaidya teaches the product wherein the cached DOM representation comprises the node and the at least one attribute is an attribute of the node (See Para 0055 and 0065). Vaidya teaches state attributes as well as data attributes at each node (See Para 0047)

With respect to **dependent claims 7-8,** as indicated in the above discussion, Vaidya in view of Lewallen teaches every element of claim 3 and 4.

Vaidya teaches access a sub-tree of a data set that is stored as a DOM tree that is rendered on the client. A DOM tree is a hierarchical tree with UI elements. (See Para 0047-0048 and 0073-0075). Vaidya teaches accessing the tree to change the font element of a given node, which is an example of accessing the sub-tree of the DOM tree.

Vaidya does not expressly teaches the product wherein the cached DOM representation comprises a sub-hierarchy of the DOM hierarchy and the node is the root node of the sub-hierarchy and the at least one attribute is an attribute of a sub-node of the sub-hierarchy where the a delta renderer to the DOM hierarchy, the delta renderer being a set of update functions that can modify the DOM representation of a UI element directly in the browser However, this limitation would have been obvious to one of ordinary skill in the art at the time of the invention, in view of Lewallen, because Lewallen teaches a change rendering process that is executed in the client (See column 16, lines 1-7).

In regard to **Independent claim 9,** Vaidya teaches a computer program product, tangibly embodied in an information carrier, the computer program product comprising instructions operable to cause data processing apparatus to perform operations comprising:

- receiving a request to render a user interface (UI) element of a UI-tree UI tree in a client computer, the UI tree representing in the client computer a hierarchical structure of UI elements contained in a user interface page received in the client computer from a server computer, the UI element being an element of the user interface page (See Para 33). Vaidya teaches the user requests to receive a UI interface that is a DOM tree. The DOM tree contains user interface elements and the DOM is received at the client from the server.

- Vaidya teaches the system renders the UI tree (See Figure 2).
- Vaidya teaches the setting of a state value that signifies that the value of the node has changed in the tree and starts a process to update the cache on the server (See Para 0058-0063), which can be considered a dirty flag.

Vaidya does not expressly teach:

- Determining whether a dirty flag is set for a cached document object model (DOM) representation of a flagged UI element that is on the path of the UI element in the UI tree,~ the flagged UI element having a reference to a node of the DOM hierarchy, the cached DOM representation being in the client computer, the UI element being different from the flagged UI element;
  if the dirty flag is set, generating in the client computer a new DOM representation of the flagged UI element and inserting the new DOM representation into the DOM hierarchy by using
- The UI tree and the reference to the node; and else, inserting the cached DOM representation in the client computer into the DOM hierarchy by using the UI tree and the reference to the node.

However, Lewallen teaches a process of accepting and embedding during runtime mixed program statements from a variety of program types at a local client by manipulating the DOM (See column 10, lines 43-67) for the purposes of modifying HTML upon the occurrence of certain types of user input. The client uses a local bridge to access APIs to modify the user interface elements that are rendered on the display. The bridge creates a new interface object and a node for the object which includes a pointed to the object that is changing (See column 7, lines 60-67 and column 8, lines 1-15 and column 11, lines 1-15). Lewallen specifically teaches a process of applying a change (delta renderer) to a document when a condition of a change is to be applied to an interface Element. For example, Lewallen teaches the process of querying the pointer to an interface object and then determining which API to call for the element then creating an element with a tag name and node name. A pointer to the new element is logged at the stub, which can be considered a flag signifying that the element is different from when it was first sent to the client. The bridge passes the pointer to the document stub. If the stub reports that the interface element is not supported by the browser then bridge looks for other objects to consider to find a match to the object type. When a match is found then the mixed program statement is executed at runtime and within the browser to render the object (See column 12, lines 44-67 and 1-67). Lewallen further teaches the process of creating a new element by referencing the embedded element in the DOM and a new element is embedded in the DOM with the correct object type (See column 14, lines 29-67). Lewallen and Vaidya teach processes of updating nodes in a document object model. They both teach processes of accessing a given node and element.

Accordingly, it would have been obvious to one of ordinary skill in the art at the time of the invention having the teachings of Vaidya and Lewallen in front of them to modify the system of Vaidya to include the client side manipulation of the node elements by storing a new element as a node in the DOM. The motivation to combine Lewallen with Vaidya comes from the suggestion in Lewallen that many interface programs, such as browsers, use and implement the W3C DOM standard and API's and by providing a mapping between the DOM standard and application object the objects can be manipulated by modifying the DOM (See column 3, lines 50-67 and column 5, lines 10-25).

With respect to **dependent claim 10,** Vaidya teaches the product wherein the cached DOM representation is a cached sub-tree of the DOM hierarchy and the node is the root node of the cached sub-tree (See Para 0065).

With respect to **dependent claim 11,** Vaidya teaches the product wherein the new DOM representation is a new sub-tree of the DOM hierarchy and the node is the root node of the new sub-tree (Se Para 0047-0048 and Para 0069-0076).

In regard to **claims 12,14,** claims 12, 14 reflects the apparatus-comprising computer readable instructions used for performing the steps as claimed in claims 1, 3 respectively, and are rejected along the same rationale.

In regard to **claim 15,** claim 15 reflects the apparatus comprising computer readable instructions used for performing the steps as claimed in claims 9 respectively, and is rejected along the same rationale.

It is noted that any citation to specific pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. In re *Heck*, 699 F.2d 1331, 1332-33,216 USPQ 1038, 1039 (Fed. Cir. 1983) (quoting In re *Lemelson*, 397 F.2d 1006,1009, 158 USPQ 275, 277 (CCPA 1968)).

Applicant's arguments with respect to claims11/12/2007 have been considered but are moot in view of the new ground(s) of rejection.

However, in the spirit of forwarding prosecution Applicant's arguments filed 11/12/2007 have been fully considered but they are not persuasive. *Applicant argues the Vaidya does not teach operations on a DOM tree to change an interface element*

Applicant argues that the teachings of Vaidya do not teach operations on a DOM UI element (See arguments page 10).

The Examiner disagrees.

The data set, while it is data, it is represented as a DOM tree for the purpose of presentation to the user whereby each of the items in the dataset are UI elements. A character on a screen is a UI element, as it is a part of the user interface. Further, the system would have to **determine** if an element has changed in order to check to see if the default attribute is set. Vaidya teaches that if the default is set then the element has not been changed.

Regarding the remaining arguments, they are directed to the limitations added with the amendment and are addressed under a new grounds of rejection

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Steven B. Theriault whose telephone number is (571) 272-5867. The examiner can normally be reached on M, W, F 10:00AM - 8:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on (571) 272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Steven B Theriault/
Patent Examiner
Art Unit 2179

WEILUN LO
SUPERVISORY PATENT EXAMINER